

Diagrama de Classes

UniSenac campus Pelotas
Profª Bruna Ribeiro
email: brgribeiro@senacrs.com.br

Diagrama de Classes

- ❑ Diagrama mais utilizado da UML
- ❑ Estrutura estática
- ❑ **Representa** os tipos (**classes**) de objetos de um sistema
 - ❑ Exibe Classes, Atributos, Operações e relações entre eles
 - ❑ Descrevem as responsabilidades do sistema

Diagrama de Classes

- ❑ Pode ser mapeado diretamente para uma linguagem O.O
- ❑ Ajuda no processo transitório dos requisitos para o código
- ❑ São utilizados para descrever, visualizar e documentar sistemas
- ❑ Pode representar visualmente o código do sistema

Diagrama de Classes

- ❑ É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos.
- ❑ Representada por um retângulo que pode possuir até três divisões:
 - ❑ Nome da classe
 - ❑ Atributos da classe
 - ❑ Métodos da classe



Relembrando

☐ Atributo

- ☐ Representa características de uma classe
- ☐ Exemplo: Jogador (nome, sexo, idade etc.).

☐ Método

- ☐ Representa atividades que um objeto de uma classe pode executar.
- ☐ Exemplo: Jogador (correr, driblar, chutar).

☐ Visibilidade

- ☐ Indica o nível de acessibilidade de um atributo ou método
- ☐ Tipos: Pública (+), Privada (-) e Protegida (#).

Diagrama de Classes

- ❑ Nome da classe
 - ❑ Localizado no compartimento superior
 - ❑ Escrito em **negrito**
 - ❑ Deve iniciar com letra maiúscula
 - ❑ Classe abstrata deve ser escrita em *itálico*.



Diagrama de Classes

❑ Atributos

- ❑ Permitem a identificação de cada objeto de uma classe
- ❑ Atributos devem conter o tipo de dados a ser armazenado
 - ❑ boolean, int, double, char, String, etc.

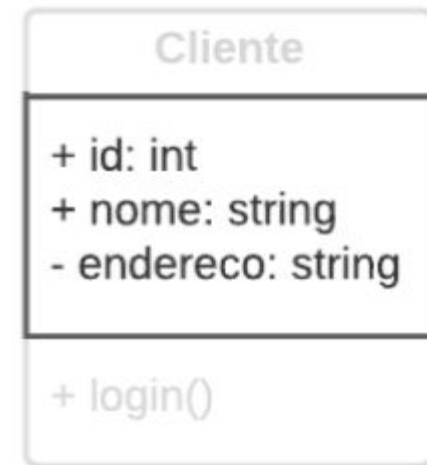


Diagrama de Classes

❑ Métodos

- ❑ Funções ou operações que uma classe pode executar.
- ❑ Eles são listados na parte inferior da classe e incluem um nome, parâmetros, tipo de retorno e visibilidade.



Diagrama de Classes

☐ Visibilidade

- ☐ Público (+) : O atributo ou método pode ser utilizado por qualquer classe
- ☐ Protegido (#) : Pode ser visto apenas pelas operações da própria classe e por suas classes herdeiras
- ☐ Privado (-) : o que pode ser visto apenas pelas operações da própria classe

Diagrama de Classes

☐ Relacionamento

☐ Os relacionamentos descrevem as associações entre as classes. Os tipos comuns de relacionamentos incluem:

- ☐ Associação
- ☐ Agregação
- ☐ Composição
- ☐ Herança (Especialização/Generalização)

Relacionamento

❑ Associação

- ❑ Descreve um vínculo que ocorre normalmente entre duas classes
- ❑ Representam o equivalente mais próximo dos relacionamentos utilizados no modelo ER
- ❑ Uma associação é representada por uma linha sólida conectando duas classes.



Relacionamento

❑ Associação - Indicadores de multiplicidade

- 1 Exatamente um
- 1..* Um ou mais
- 0..* Zero ou mais (muitos)
- * Zero ou mais (muitos)
- 0..1 Zero ou um
- m..n Faixa de valores (por exemplo: 4..7)

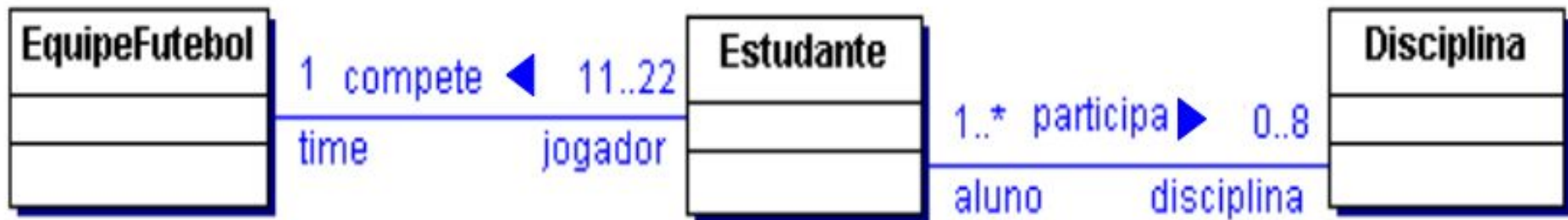


Relacionamento

- ❑ Associação - Exemplo
 - ❑ Um **Estudante** pode ser
 - ❑ Um aluno de uma **Disciplina** e
 - ❑ um jogador da **Equipe de Futebol**
 - ❑ Cada **Disciplina** deve ser cursada por no mínimo 1 aluno
 - ❑ Um aluno pode cursar de 0 até 8 disciplinas

Relacionamento

- ❑ Associação - Exemplo
 - ❑ Um **Estudante** pode ser
 - ❑ Um aluno de uma **Disciplina** e
 - ❑ um jogador da **Equipe de Futebol**
 - ❑ Cada **Disciplina** deve ser cursada por no mínimo 1 aluno
 - ❑ Um aluno pode cursar de 0 até 8 disciplinas



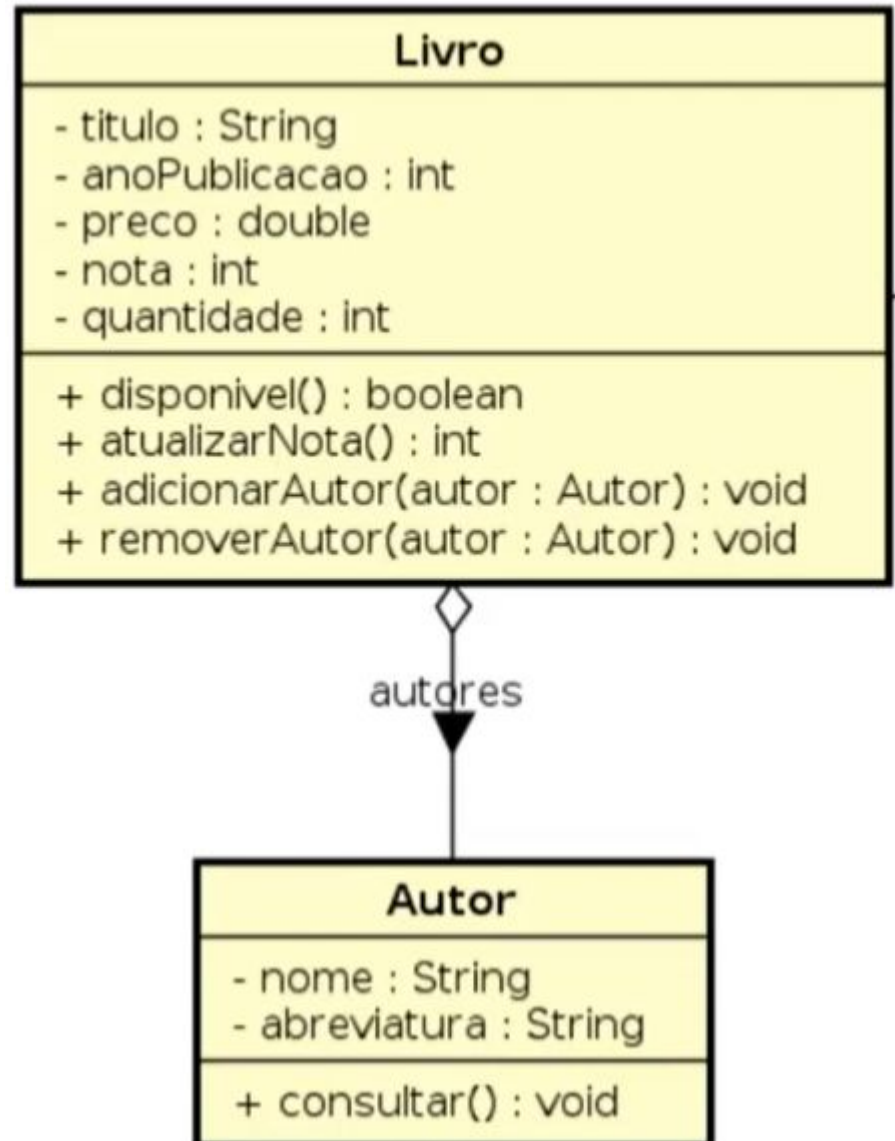
Relacionamento

- ❑ Agregação
 - ❑ Tipo especial de associação que tenta demonstrar que as informações de um objeto-todo precisam ser complementadas pelas informações contidas em um (ou mais) objetos-parte.
 - ❑ A existência do **objeto-parte** faz sentido mesmo não existindo o **objeto-todo**.

Relacionamento

□ Agregação

□ Exemplo



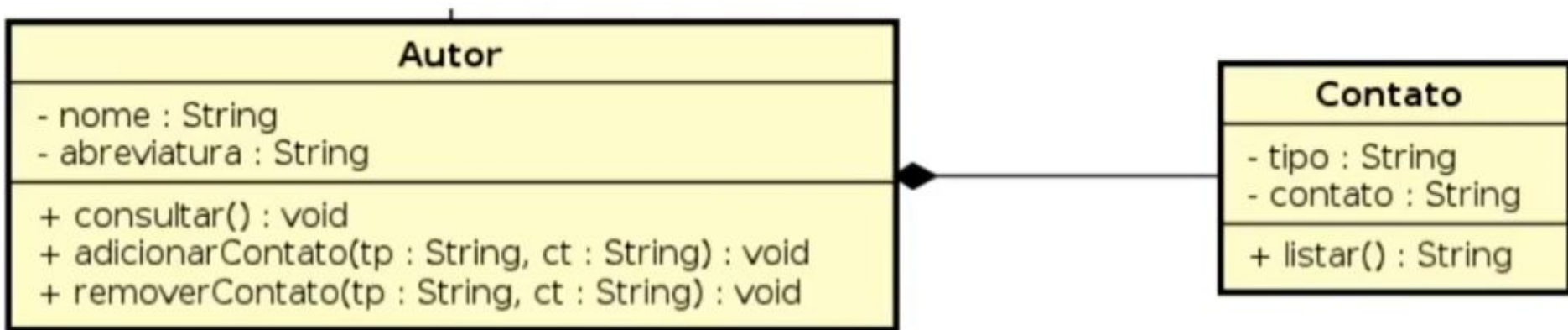
Relacionamento

- ❑ Composição
 - ❑ É uma variação da agregação e considerada mais “forte”
 - ❑ O **objeto-parte** não pode existir sem o **objeto-todo**.
 - ❑ Se o **objeto-todo** for destruído, o **objeto-parte** também será.

Relacionamento

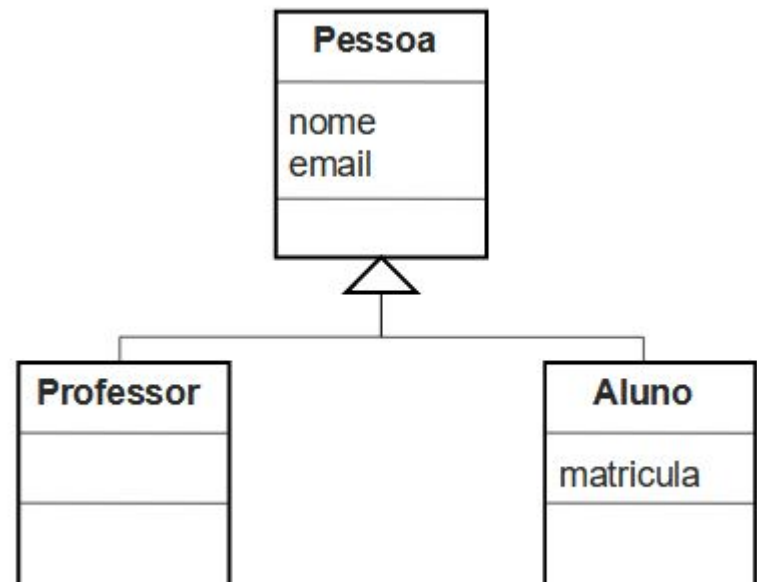
❑ Composição

❑ Exemplo



Relacionamento

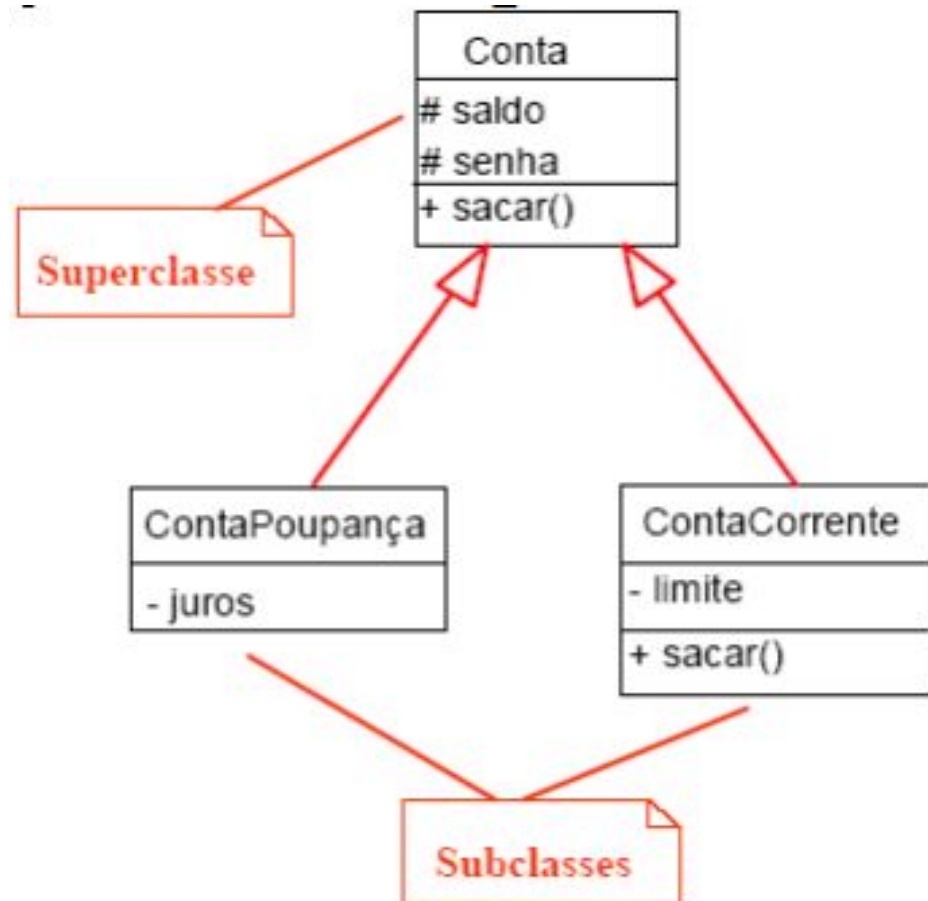
- ❑ Especialização/Generalização
 - ❑ Identificar classes-mãe (gerais) e classes-filhas (especializadas)
 - ❑ Atributos e métodos definidos na **classe-mãe** são herdados pelas **classes-filhas**



Relacionamento

□ Especialização/Generalização

□ Exemplo

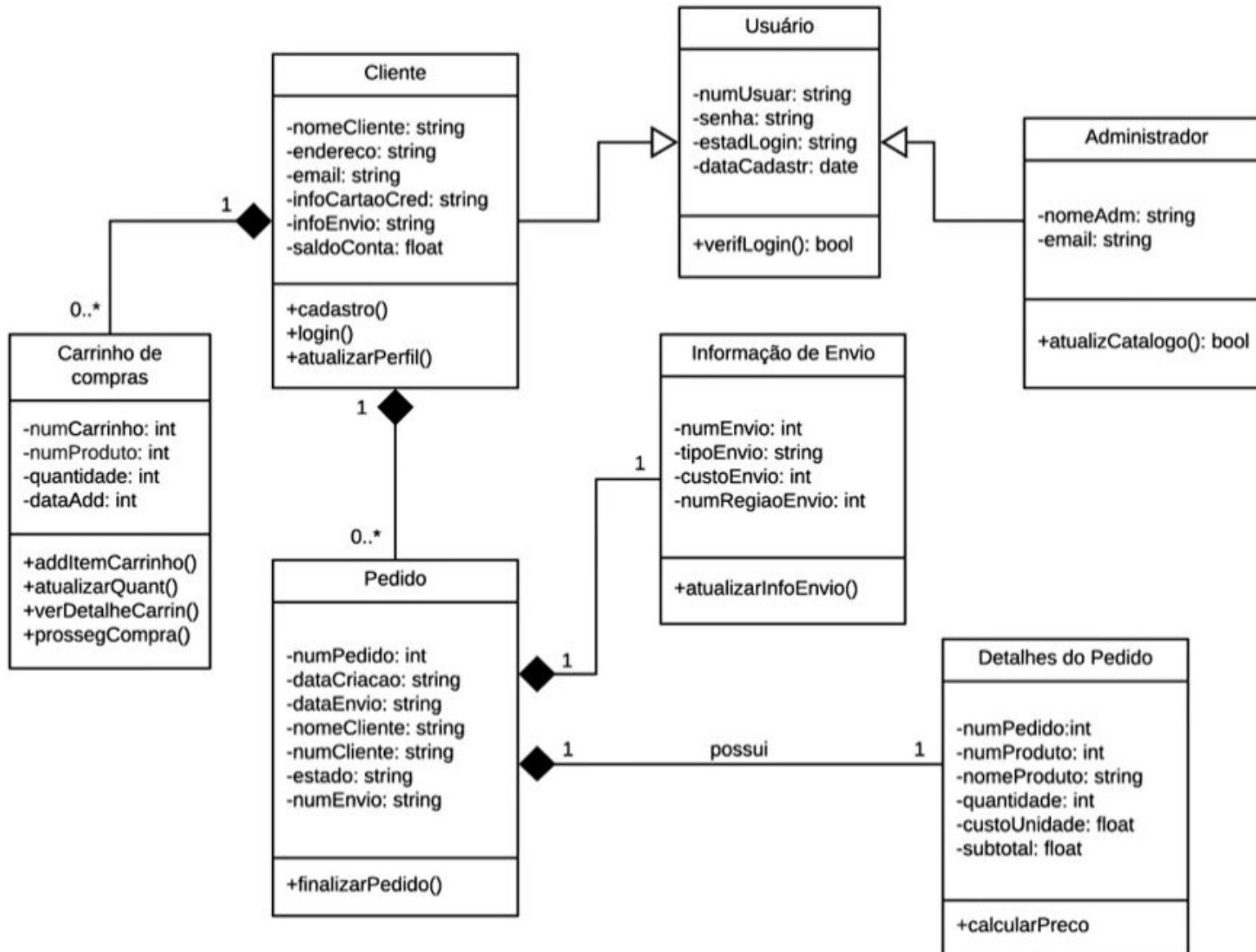


Dúvidas

❏ Dúvidas????



Exemplo



Unisenac
Centro Universitário RS



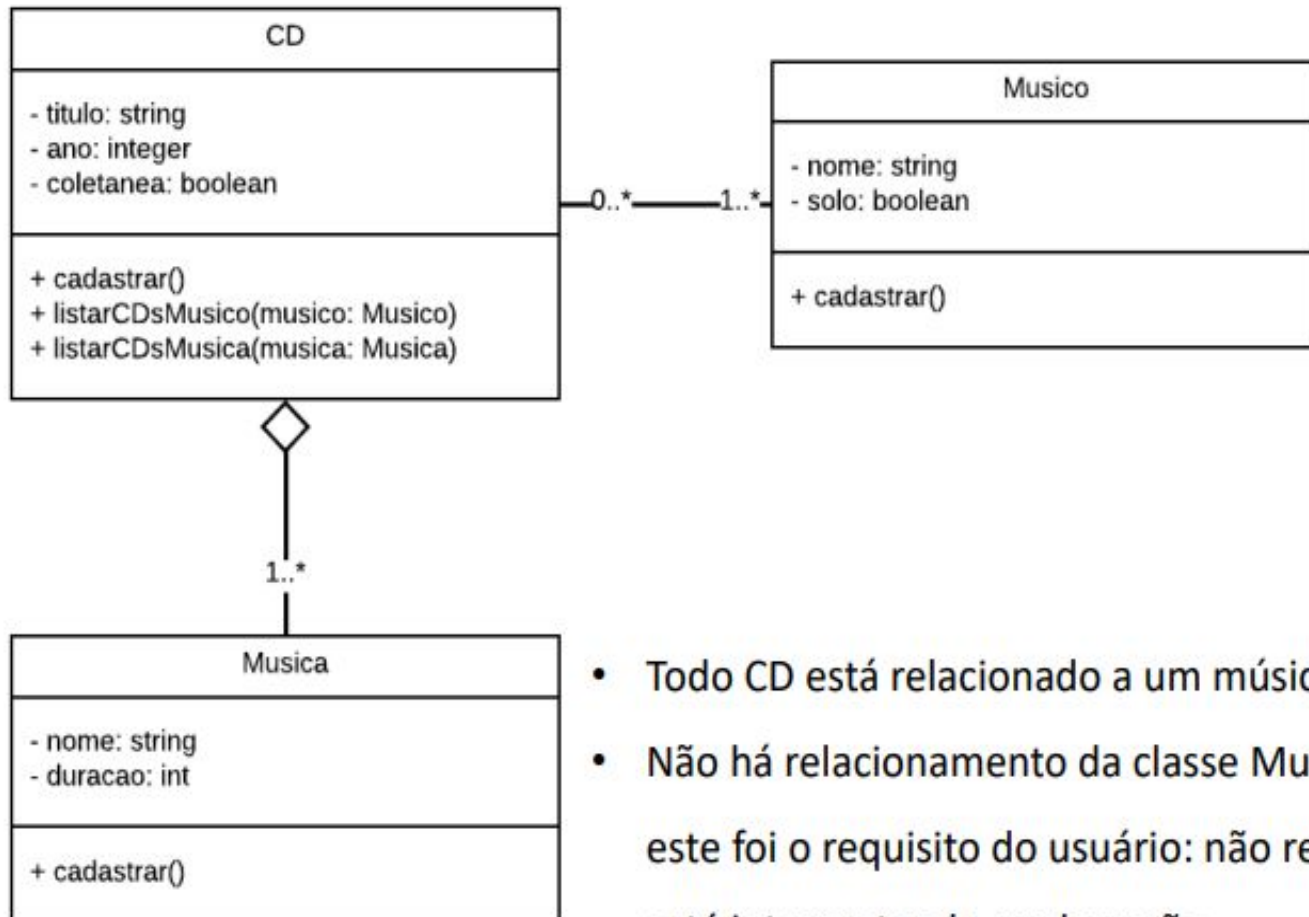
Exemplo

Exemplo

- ☐ CD's do Angelo
 - ☐ Angelo tem uma coleção grande de CD's e gostaria de cadastrar no seu computador a lista desses CD's, pois às vezes nem sabe o que tem.
 - ☐ Ele pensou em cadastrar o nome do(a) cantor(a) ou conjunto, o título do CD e o ano de lançamento.
 - ☐ Angelo notou que alguns CD's são de coletâneas.
 - ☐ Sendo assim, não têm apenas um(a) cantor(a) (ou conjunto) e sim vários(as).
 - ☐ Ele quer cadastrar essa lista de músicos, sem relacioná-los às músicas.
 - ☐ Deseja controlar também se o CD é de coletânea.
 - ☐ Angelo gostaria de ter cadastrada a lista das músicas de cada CD, com o tempo de duração de cada faixa.
 - ☐ São relatórios desejados: Os CD's de um determinado músico e em quais CD's está uma determinada música

Exemplo

CD's do Angelo



- Todo CD está relacionado a um músico
- Não há relacionamento da classe Musica com a classe Musico, pois este foi o requisito do usuário: não realizar controle de que músico está interpretando qual canção.